Automatic Trimap Generation for Image Matting

Vikas Gupta
Shri Ramdeobaba College of Engineering and Management
Nagpur, India
Email: guptavr1@rknec.edu

Shanmuganathan Raman
Indian Institute of Technology Gandhinagar
Gandhinagar, India
Email: shanmuga@iitgn.ac.in

***Abstract*—Image matting is an important problem in computational photography. Although, it has been studied for more than two decades, yet there is a challenge of developing an automatic matting algorithm which does not require any human intervention. Most of the state-of-the-art matting algorithms require human intervention in the form of trimap or scribbles to generate the alpha matte form the input image. In this paper, we present a simple and efficient approach to automatically generate the trimap from the input image and make the whole matting process free from human-in-the-loop. We use learning based matting method to generate the matte using the automatically generated trimap. Experimental results demonstrate that our method produces good quality trimap which results into accurate matte estimation. We validate our results by replacing the automatically generated trimap by manually created trimap while using the same image matting algorithm.**

I. Introduction

Image matting is the process of accurately estimating the foreground object in images and videos. It is a very important technique in image and video editing applications, particularly in film production for creating visual effects. In image matting process some pixels may have contribution from foreground as well as background, such pixels are called *partial* or *mixed* pixels.

Given an image I, the image matting problem is mathematically stated as given in Eq. 1.

$$I_p = \alpha_p F_p + (1 - \alpha_p) B_p \,. \tag{1}$$

Where, $p = (x, y)$, α_p represents the matte and it can take any value in $[0, 1]$, and F_p and B_p are foreground and background pixel value respectively. If $\alpha_p = 1$ or 0 then the pixel at location p belongs to *definite foreground* or *defiinite background* respectively. Otherwise that pixel is called a *partial* or a *mixed* pixel. In order to fully separate the foreground from the background in an image, accurate estimation of the alpha values for partial or mixed pixels is necessary. In Eq. 1, if we consider a full color image (RGB), there are 7 unknowns (F_p, B_p for each color channel and α_p) and three equations (one for each color channel). Thus image matting is a severely under-constrained problem. Such under-constrained problems can be solved by adding more information into them. This additional information is provided in the form of *trimap* [1] or *scribbles* [2], *i.e.*, labeling some pixels belonging to definite foreground or definite background.

In order to fully extract meaningful foreground object, almost all the matting techniques rely on the user intervention, wherein the user segments the input image into three regions: definite foreground, definite background, and unknown region. This three-level map is called a *trimap*. Ideally, the trimap should consist of very small unknown region around the foreground boundary and it should contain only the partial or mixed pixels. The smaller the unknown region (less number of mixed pixels), the more accurate will be the estimated matte. However generating such an accurate trimap requires lot of human efforts and it is often undesirable, particularly in the case of transparent objects. Thus, accuracy of a trimap is one of the important factor which affects the performance of a matting algorithm [**?**]. Therefore, to alleviate such problems user specified *trimap* or *scribbles* are needed to get the highly accurate matte. However, we can reduce the user efforts for manually creating the trimap by automatically generating more accurate trimap.

In this paper, we propose a novel method to automatically generate trimap from the given image. We use the saliency map of the image to generate the trimap. First, we oversegment the image using SLIC superpixel algorithm [3]. Then we obtain the local features using *Oriented Texture Curves* (OTC) feature descriptor for each superpixel in the over-segmented image [4]. These feature vectors are then clustered to obtain the background and foreground superpixels. Then we update the saliency map of the image and threshold it to obtain the binary map. This binary map is then eroded and dilated in order to obtain the desired trimap. The steps involved in the proposed method is depicted in Fig. 1. The main contributions of our paper are given below.

1) We propose an automatic trimap generation framework for image matting to get rid of any human intervention.
2) Instead of working on each pixel, we employ superpixels to over-segment the image and process a group of pixels together.
3) We use image saliency and an appropriate local feature descriptor to identify the foreground and background superpixels which helps in automatic generation of trimap.

The rest of the paper is organized as follows. In section II, we briefly survey the existing state-of-the-art matting algorithms. Section III gives the details of the proposed automatic trimap generation algorithm. In section IV, we show and discuss the results of image matting obtained using the trimap generated from our approach. Section V concludes the paper with some ideas for future improvements.

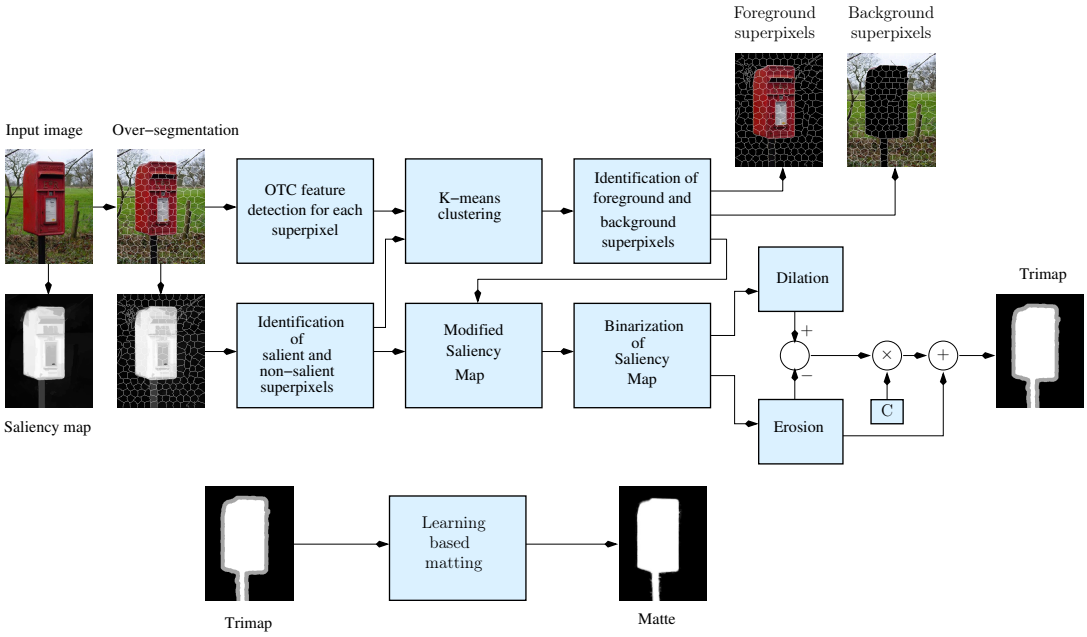

Fig. 1. Proposed saliency based automatic trimap generation framework.

II. Related Work

In this section, we discuss some of the recent state-of-the-art matting algorithms. Generally the matting algorithms are classified as *sampling based approaches* [5], [1], [2] and *affinity based approaches* [6], [7], [8], [9], [10].

A. Sampling based approaches

The basic principle of these approaches is to use neighboring foreground and background pixels as samples to estimate the alpha values for the unknown pixels. Ruzon and Tomasi proposed a *sampling* based approach for matting [5]. In this approach, alpha values are measured along a manifold which connects the color distribution frontier of each object. The unknown region is divided into subregions and a local window is defined in these subregions such that it covers the unknown region, and a local foreground and background region. The optimal alpha is responsible for an intermediate distribution which has maximum probability for observed color values.
The *Bayesian approach* proposed by Chuang *et al*. also uses the probabilistic approach to solve the matting problem [1]. The matting problem is formulated using a Bayesian framework and maximum a posteriori (MAP) technique is used to solve for the matte.
The previous two methods assume that the unknown region is somewhat narrow around the foreground boundary and therefore they use local color models. But this assumption fails if the trimap is not well defined and it consist of only a few scribbles. In the case of rough trimap, global sampling method is used to tackle the sampling problem [2]. The sampling based approaches works well when the trimap is well defined.

B. Affinity based approaches

The affinity based approaches utilize the local image statistics by defining various *affinities* between neighboring pixels to model the matte gradient across the image instead of directly estimating the alpha value at each pixel. *Poisson matting* estimates the matte gradient from the image using boundary information from a user-supplied trimap and then reconstructs the matte by solving Poisson equation [6]. It is based on the assumption that intensity change in the foreground and the background is smooth. Grady *et al*. employed *random walk* algorithm to calculate the final alpha values based on affinity [7]. The *geodesic* matting method measures the weighted geodesic distance from the user-provided scribbles to the pixels in the unknown region (outside of the scribbles) for labeling them as foreground or background pixel [8].
Zheng *et al*. proposed an *interactive matting* algorithm which is similar to geodesic matting called *FuzzyMatte* [9]. This method computes the *fuzzy connectedness* between the unknown pixel and the known foreground and background pixels. The final alpha value is then calculated using the fuzzy connectedness. The Closed-form matting approach explicitly derives a cost function from local smoothness assumptions on foreground and background colors [10]. This cost function can be solved by a sparse linear system of equations, which yields the globally optimal alpha matte. The affinity used in this approach does not have any global parameters. Instead, it uses local estimates of mean and variances which leads to significant improvement in the performance as demonstrated in [10].

C. Other approaches

Robust matting method combines the color sampling and affinity together in a single optimization process to get more

accurate and robust matting solution [11]. It samples the foreground and background colors for unknown pixels and determines the confidence of these samples. The high confidence samples are made to contribute to the matting energy function which is minimized using a random walk. Zheng and Kambhamettu utilized semi-supervised learning to solve the digital matting problem which results in a local learning based matting approach and a global learning based approach [12]. We use this image matting algorithm to evaluate the effectiveness of the automatic trimaps generated in this work.

III. Automatic Trimap Generation

In this section, we describe in detail our proposed framework for automatically generating the *trimap* from a given image. We assume that there is a single salient object present in the given scene. The complete framework is divided into three parts as: over-segmentation and feature description, identification of background and foreground superpixels, and trimap generation and matting.

A. Over-segmentation and feature description

Consider an input image I as shown in Fig. 2(a). We segment the image I into N superpixels using the algorithm given in [3]. The resulting over-segmented image is shown in Fig. 2(b). Note that each superpixel contains distinct texture and color information, therefore we compute the OTC features for a patch of size 13×13 in each superpixel [4]. We use the approach similar to the one proposed in [13] for extracting a patch from a superpixel. The OTC descriptor captures the texture of a patch along multiple orientations. This descriptor is shown to be robust to illumination changes, geometric distortions, and local contrast differences. It provides a 185-dimensional texture feature in eight different directions.

We obtain the saliency maps SM_i, for $i \in \{1, 2, 3\}$, of the input image using three different methods [14], [15], [16]. Each of these methods uses different framework to obtain the saliency map. In [14], Huaizu Jiang *et al*. employed supervised learning approach to integrate regional features such as the regional contrast, regional property, and regional backgroundness descriptors together to form the master saliency map. In [15], the image is segmented to obtain a set of object candidates and then a fixation algorithm is used to rank different regions based on their saliency score. In [16], Rui Zhao *et al*. utilized global context and local context models to obtain multi-context saliency model using convolutional neural networks (CNN). The saliency maps SM_i, for $i \in \{1, 2, 3\}$, obtained from these three methods are then combined to get a single saliency map (see Fig. 2(c)) as given in Eq. 2.

$$\text{SM} = \beta_1 \times \text{SM}_1 + \beta_2 \times \text{SM}_2 + \beta_3 \times \text{SM}_3 \,. \tag{2}$$

where, β_1, β_2, and β_3 are non-negative constants. We choose the same value of $\frac{1}{3}$ for β_1, β_2, and β_3 in this work.

B. Identification of background and foreground superpixels

We use the saliency map SM to classify superpixels into salient and non-salient superpixels. For each superpixel, we obtain the median intensity value in the saliency map. If this median value is greater than a threshold T_1 then that superpixel is classified as a salient superpixel. Otherwise, it is classified as a non-salient superpixel. Initially, we consider the salient superpixels as foreground superpixels and non-salient superpixels as background superpixels. It may happen that some salient superpixels may be mis-classified as background and some non-salient superpixels may be mis-classified as foreground. To alleviate this problem, we cluster the OTC features of superpixels initially classified as foreground into five different clusters using k-means clustering [17]. Similarly we cluster the OTC features of superpixels initially classified as background into five different clusters using k-means clustering.

For each superpixel, which was initially classified as foreground, we compute the Euclidean distances D_{fb} between that superpixel and the cluster centers of the background superpixels. If the minimum of the computed Euclidean distances (D_{fb}) is less than a threshold T_2 then that superpixel is identified as a background superpixel. The same process is repeated for the superpixels which were initially classified as background to identify more foreground superpixels. We repeat the same process for all the superpixels identified as background using the cluster centers estimated by the foreground superpixels. The separated foreground and background superpixels are shown in Fig. 2(d, e). Based on this information we modify the saliency map SM so that only the foreground region will have the salient value. Finally, we arrive at the modified saliency map SM' as shown in Fig. 2(f).

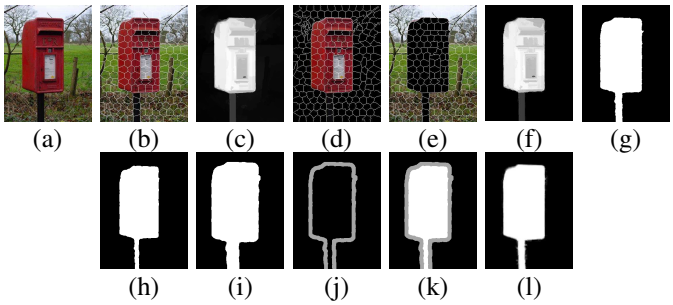

Fig. 2. Intermediate Results: (a) Input image, (b) Over-segmented image, (c) Saliency map, (d) Foreground superpixels, (e) Background superpixels, (f) Modified saliency map, (g) Binarized saliency map, (h) Eroded saliency map, (i) Dilated saliency map, (j) Difference map, (k) Trimap, (l) Estimated matte using [12].

C. Trimap generation and matting

To generate the *trimap*, we need a binarized saliency map. The modified saliency map SM' is binarized using Otsu's thresholding method [18] and the resulting binarized saliency map is shown in Fig. 2(g). The binarized saliency map is then eroded and dilated to get the eroded map SM_e and the dilated map SM_d as shown in Fig. 2(h, i). We use two disk structuring elements of radii 5 and 10 for the erosion and dilation operations, respectively. The eroded map SM_e

is subtracted from the dilated map SM_d to get the unknown region of the trimap as given in Eq. 3.

$$\text{SM}_{diff} = \text{SM}_d - \text{SM}_e \ . \tag{3}$$

The obtained difference map is multiplied with a constant C, where $0 < C < 1$ (see Fig. 2(j)). This difference map is then added to the eroded saliency map SM_e, which results in a *trimap* (TM) as shown in Fig. 2(k). This process is explained in Eq. 4.

$$\text{TM} = C \times \text{SM}_{diff} + \text{SM}_e \ . \tag{4}$$

We use the *Learning based matting* technique, proposed in [12], to obtain the alpha matte for the input image I by using the *trimap* obtained from our proposed framework. The estimated alpha matte is depicted in the Fig. 2(l).

IV. Results and Discussion

In this section, we present and discuss the results obtained by the proposed framework. We test our proposed method on a number of images obtained from FT [19] and PASCAL-S [15] datasets. We compare the trimaps generated by the proposed framework with the manually created trimaps. The proposed method works well in the case of images where the background part is natural, which can be noticed in Fig. 3. The first column shows the input images, the second column depicts the manually created trimaps, in the third column the trimaps generated by the proposed approach are shown. The mattes corresponding to both these trimaps are shown in the fourth and the fifth column respectively.

The first row of Fig. 3 shows the results for an image which consists of a foreground object (A hollow box) and a natural background. Here, we observe that the automatically generated trimap is quite similar to that of manually created trimap thereby leading to accurate matte estimation. Similar observation can be made for the images shown in the Third, fourth, and fifth rows. For the image shown in the second row, there is little difference in the automatically generate trimap and the manually created trimap. Some part of the foreground is marked as unknown in the automatically generated trimap, which is marked as definite foreground in the manually created trimap. However, the matting algorithms takes care of it and we get approximately similar mattes from both these trimaps. In the fifth row, we can notice that the trimap obtained using the proposed approach marks the unknown region (foreground boundary) very accurately compared to that of the manually created trimap.

The results illustrated in Fig. 3 demonstrate that the automatically generated trimap is as accurate as the manually created trimap for generating the mattes. To validate our claim, we compute sum of squared differences (SSD) for the matte generated using two different trimaps *i.e.*, trimap using the proposed approach and the manually created trimap. The SSD for the images in the first to the seventh row are $106, 92, 23, 38,$ and 58, respectively. We observed that the SSD values are very small. The proposed method has some limitations which can be observed in the case of images in which background is synthetically generated. If there is an ambiguity between foreground and background color, then the proposed method might lead to some errors in the trimap.

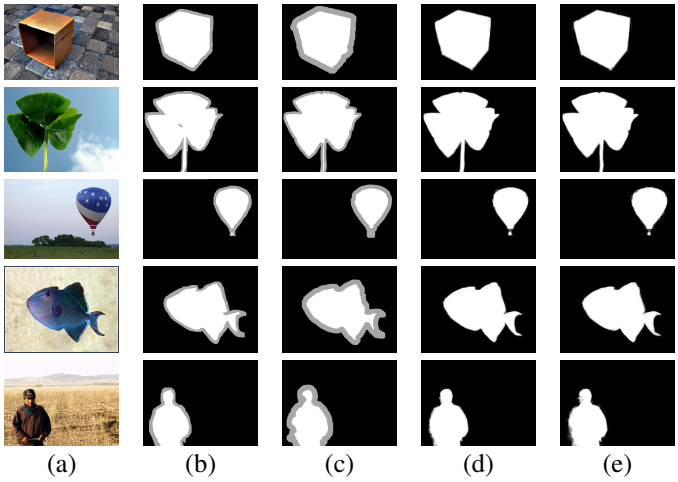

Fig. 3. (a) Input image, (b) Trimap (manually generated), (c) Trimap (Using proposed approach), (d) Matte estimation by using (b) (using [12] algorithm), (e) Matte estimation by using (c) (using [12] algorithm).

We implemented this framework in MATLAB on a PC with Intel i5-4460s 2.9 GHz processor and 12 GB RAM. For segmenting the image into superpixels we set the value of N in the range of 250 to 400. The threshold T_1 is set to be equal to 30% of the highest value in the saliency map. The threshold value T_2 is set to be equal to the mean of distances between the OTC feature vectors of superpixels belonging to foreground (or background) and the cluster centers of the background (or foreground) superpixels. The constant C is chosen to be equal to 0.65. The proposed method takes typically 10 seconds to generate the trimap for any given image thereby automating the entire image matting pipeline.

V. Conclusion

Image matting is an important process for accurate estimation of foreground object from the background in image and video editing applications. This task is ill-posed, thereby poses a significant challenge for computational photography. Almost all the matting algorithms require user intervention in the form of trimap or scribbles as input to these algorithms. The performance of these algorithms depends on such user inputs. Also manually creating a trimap consumes a lot of time. To alleviate this problem and make the whole matting process automatic, we have proposed a simple and efficient framework for automatically generating the trimap for a given input image. The experimental results demonstrate that the automatically generated trimaps are very close to that of manually created trimaps which results in accurate estimation of matte. There could be images where there is no distinct salient object present. In such a scenario, generating the trimap automatically is a challenge to be addressed in future. Another future challenge is to extract matte corresponding to multiple foreground objects from a given image automatically.

REFERENCES

[1] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A bayesian approach to digital matting," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. II–264.

[2] J. Wang and M. F. Cohen, "An iterative optimization approach for unified image segmentation and matting," in *International Conference on Computer Vision*, vol. 2. IEEE, 2005, pp. 936–943.

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, 2012.

[4] R. Margolin, L. Zelnik-Manor, and A. Tal, "Otc: A novel local descriptor for scene classification," in *ECCV*. Springer, 2014, pp. 377–391.

[5] M. A. Ruzon and C. Tomasi, "Alpha estimation in natural images," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 18–25.

[6] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, "Poisson matting," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 315–321, 2004.

[7] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, "Random walks for interactive alpha-matting," in *Proceedings of VIIP*, 2005, pp. 423–429.

[8] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[9] Y. Zheng, C. Kambhamettu, J. Yu, T. Bauer, and K. Steiner, "Fuzzymatte: A computationally efficient scheme for interactive matting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[10] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 228–242, 2008.

[11] J. Wang and M. F. Cohen, "Optimized color sampling for robust matting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[12] Y. Zheng and C. Kambhamettu, "Learning based digital matting," in *International Conference on Computer Vision*, 2009, pp. 889–896.

[13] J. Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels," in *ECCV*. Springer, 2010, pp. 352–365.

[14] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 2083–2090.

[15] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille, "The secrets of salient object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 280–287.

[16] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1265–1274.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[19] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1597–1604.